YOLOP: You Only Look Once for Panoptic Driving Perception

Dong Wu Manwen Liao Weitian Zhang Xinggang Wang
Xiang Bai Wenqing Cheng Wenyu Liu
School of EIC, Huazhong University of Science & Technology
{riserwu,mwliao,wtzhang,xgwang,xbai,chengwq,liuwy}@hust.edu.cn

Abstract

A panoptic driving perception system is an essential part of autonomous driving. A high-precision and real-time perception system can assist the vehicle in making the reasonable decision while driving. We present a panoptic driving perception network (YOLOP) to perform traffic object detection, drivable area segmentation and lane detection simultaneously. It is composed of one encoder for feature extraction and three decoders to handle the specific tasks. Our model performs extremely well on the challenging BDD100K dataset, achieving state-of-the-art on all three tasks in terms of accuracy and speed. Besides, we verify the effectiveness of our multi-task learning model for joint training via ablative studies. To our best knowledge, this is the first work that can process these three visual perception tasks simultaneously in real-time on an embedded device Jetson TX2(23 FPS) and maintain excellent accuracy. To facilitate further research, the source codes and pre-trained models are released at https://github.com/hustvl/YOLOP

1. Introduction

Recently, extensive research on autonomous driving has revealed the importance of panoptic driving perception system. It plays a significant role in autonomous driving as it can extract visual information from the images taken by the camera and assist the decision system to control the actions of the vehicle. In order to restrict the maneuver of vehicles, the visual perception system should be able to understand the scene and then provide the decision system with information including: locations of the obstacles, judgements of whether the road is drivable, the position of the lanes etc. Object detection is usually involved in the panoptic driving perception system to help the vehicles avoid obstacles and follow traffic rules. Drivable area segmentation and lane detection are also needed as they are crucial for planning the driving route of the vehicle.

For such a panoptic driving perception system, high-precision and real-time are two most critical requirements, which are related to whether the autonomous vehicle can make accurate and timely decision to ensure safety. However, for practical autonomous driving system, especially the ADAS, the computational resources are often marginal and limited. Therefore, it is very challenging to take both requirements into account in real-world scenarios.

(a) Input

(b) Output

Figure 1. The input and output of our model. The purpose of our model is to process traffic objects detection, drivable area segmentation and lane detection simultaneously in one input image. In (b), the brown bounding boxes indicate traffic objects, the green areas are the drivable areas, and the blue lines represent the lane line.

Many methods handle these tasks separately. For instance, Faster R-CNN [24] and YOLOv4 [1] deal with object detection; ENet [19] and PSPNet [30] are proposed to perform semantic segmentation. SCNN [18] and SAD-ENet [9] are used for detecting lanes. Despite the excellent performance these methods achieve, processing these tasks one after another takes longer time than tackling them all at once. When deploying the panoptic driving perception system on embedded devices commonly used in the self-driving car, limited computational resources and latency should be taken into account. In addition, different tasks in traffic scenes understanding often have much related information. As shown in Figure 1, lanes are often the boundary of drivable area, and drivable area usually closely surrounds the traffic objects. A multi-task network is more suitable in this situation as (1) it can accelerate the image analysis process by handling multiple tasks simultaneously rather than sequentially. (2) it can share information among multiple tasks as multi-task network often shares the same feature extraction backbone. Therefore, it is of essence to explore multi-task approaches in autonomous driving.

In order to solve the multi-task problem for panoptic driving perception, i.e., traffic object detection, drivable area segmentation and lane detection, while obtaining high precision and fast speed, we design a simple and efficient network architecture. We use a lightweight CNN [26] as the encoder to extract features from the image. Then these feature maps are fed to three decoders to complete their respective tasks. Our detection decoder is based on the current best-performing single-stage detection network [1] for two main reasons: (1) The single-stage detection network is faster than the two-stage detection network. (2) The grid-based prediction mechanism of the single-stage detector is more related to the other two semantic segmentation tasks, while instance segmentation is usually combined with the region-based detector [7]. And we verify the two viewpoints in the experiments section. The feature map output by the encoder incorporates semantic features of different levels and scales, and our segmentation branch can use these feature maps to complete pixel-wise semantic prediction excellently.

In addition to the end-to-end training strategy, we attempt some alternating optimization paradigms which train our model step-by-step. On the one hand, we can put unrelated tasks in different training steps to prevent interlimitation. On the other hand, the task trained first can guide other tasks. So this kind of paradigm sometimes works well though cumbersome. However, experiments show that it is unnecessary for our model as the one trained end to end can perform well enough. Our panoptic driving perception system reaches 41 FPS on a single NVIDIA TITAN XP and 23 FPS on Jetson TX2; meanwhile, it achieves state-of-the-art on the three tasks of the BDD100K dataset [28].

In summary, our main contributions are: (1) We put forward an efficient multi-task network that can jointly handle three crucial tasks in autonomous driving: object detection, drivable area segmentation and lane detection to save computational costs and reduce inference time. Our work is the first to reach real-time on embedded devices while maintaining state-of-the-art level performance on the BDD100K dataset. (2) We design the ablative experiments to verify the effectiveness of our multi-tasking scheme. It is proved that the three tasks can be learned jointly without tedious alternating optimization. (3) We design the ablative experiments to prove that the grid-based prediction mechanism of detection task is more related to that of semantic segmentation task, which is believed to provide reference for other relevant multi-task learning research works.

2. Related Work

In this section, we review solutions to the above three tasks respectively, and then introduce some related multi-task learning work. We only concentrate on solutions based on deep learning.

2.1. Traffic Object Detection

In recent years, with the rapid development of deep learning, many prominent object detection algorithms have emerged. Current mainstream object detection algorithms can be divided into two-stage methods and one-stage methods.

Two-stage methods complete the detection task in two steps. First, regional proposals are obtained, and then features in the regional proposals are used to locate and classify the objects. The generation of regional proposals has gone through several stages of development [2, 4, 5, 24].

The SSD-series [14] and YOLO-series algorithms are milestones among one-stage methods. This kind of algorithm performs bounding box regression and object classification simultaneously. YOLO [21] divides the picture into S×S grids instead of extracting regional proposals with the RPN network, which significantly accelerates the detection speed. YOLO9000 [22] introduces the anchor mechanism to improve the recall of detection. YOLOv3 [23] uses the feature pyramid network structure to achieve multi-scale detection. YOLOv4 [1] further improves the detection performance by refining the network structure, activation function, loss function and applying abundant data augmentation.

2.2. Drivable Area Segmentation

Due to the rapid development of deep learning, a number of CNN-based methods have made great success in semantic segmentation area, and they can be applied in drivable area segmentation task to provide pixel-level results. FCN [15] firstly introduces fully convolutional network to

semantic segmentation. Despite the skip-connection refinement, its performance is still limited by low-resolution output. PSPNet [30] comes up with the pyramid pooling module to extract features in various scales to enhance its performance. Besides accuracy, speed is also a key element in evaluating this task. In order to achieve real-time inference speed, ENet [19] reduces size of the feature maps. Recently, multitask learning is introduced to deal with this task, EdgeNet [6] combine edge detection with drivable area segmentation task to obtain more accurate segmentation results without compromising the inference speed.

2.3. Lane Detection

In lane detection, there are lots of innovative researches based on deep learning. [17] constructs a dual-branch network to perform semantic segmentation and pixel embedding on images. It further clusters the dual-branch features to achieve lane instance segmentation. SCNN [18] proposes slice-by-slice convolution, which enables the message to pass between pixels across rows and columns in a layer, but this convolution is very time-consuming. Enet-SAD [9] uses self attention distillation method, which enables low-level feature maps to learn from high-level feature maps. This method improves the performance of the model while keeping the model lightweight.

2.4. Multi-task Approaches

The goal of multi-task learning is to learn better representations through shared information among multiple tasks. Especially, a CNN-based multitask learning method can also achieve convolutional sharing of the network structure. Mask R-CNN [7] extends Faster R-CNN by adding a branch for predicting object mask, which combines instance segmentation and object detection tasks effectively, and these two tasks can promote each other's performance. LSNet [3] summarizes object detection, instance segmentation and pose estimation as location-sensitive visual recognition and uses a unified solution to handle these tasks. With a shared encoder and three independent decoders, MultiNet [25] completes the three scene perception tasks of scene classification, object detection and segmentation of the driving area simultaneously. DLT-Net [20] inherits the encoder-decoder structure, and contributively constructs context tensors between sub-task decoders to share designate information among tasks. [29] puts forward mutually interlinked sub-structures between lane area segmentation and lane boundary detection. Meanwhile, it proposes a novel loss function to constrain the lane line to the outer contour of the lane area so that they're going to overlap geometrically. However, this prior assumption also limits its application as it only works well on scenarios where the lane line tightly wraps the lane area. What's more, the training paradigm of multitask model is also worth thinking about. [10] states that the joint training is appropriate and beneficial only when all those tasks are indeed related; otherwise, it is necessary to adopt alternating optimization. So Faster R-CNN [24] adopts a pragmatic 4-step training algorithm to learn shared features. This paradigm sometimes may be helpful, but mostly it is tedious.

3. Methodology

We put forward a simple and efficient feed-forward network that can accomplish traffic object detection, drivable area segmentation and lane detection tasks altogether. As shown in Figure 2, our panoptic driving perception single-shot network, termed as YOLOP, contains one shared encoder and three subsequent decoders to solve specific tasks. There are no complex and redundant shared blocks between different decoders, which reduces computational consumption and allows our network to be easily trained end-to-end.

3.1. Encoder

Our network shares one encoder, which is composed of a backbone network and a neck network.

3.1.1 Backbone

The backbone network is used to extract the features of the input image. Usually, some classic image classification networks serve as the backbone. Due to the excellent performance of YOLOv4 [1] on object detection, we choose CSP-Darknet [26] as the backbone, which solves the problem of gradient duplication during optimization [27]. It supports feature propagation and feature reuse which reduces the amount of parameters and calculations. Therefore, it is conducive to ensuring the real-time performance of the network.

3.1.2 Neck

The neck is used to fuse the features generated by the backbone. Our neck is mainly composed of Spatial Pyramid Pooling (SPP) module [8] and Feature Pyramid Network (FPN) module [11]. SPP generates and fuses features of different scales, and FPN fuses features at different semantic levels, making the generated features contain multiple scales and multiple semantic level information. We adopt the method of concatenation to fuse features in our work.

3.2. Decoders

The three heads in our network are specific decoders for the three tasks.

3.2.1 Detect Head

Similar to YOLOv4, we adopt an anchor-based multi-scale detection scheme. Firstly, we use a structure called Path

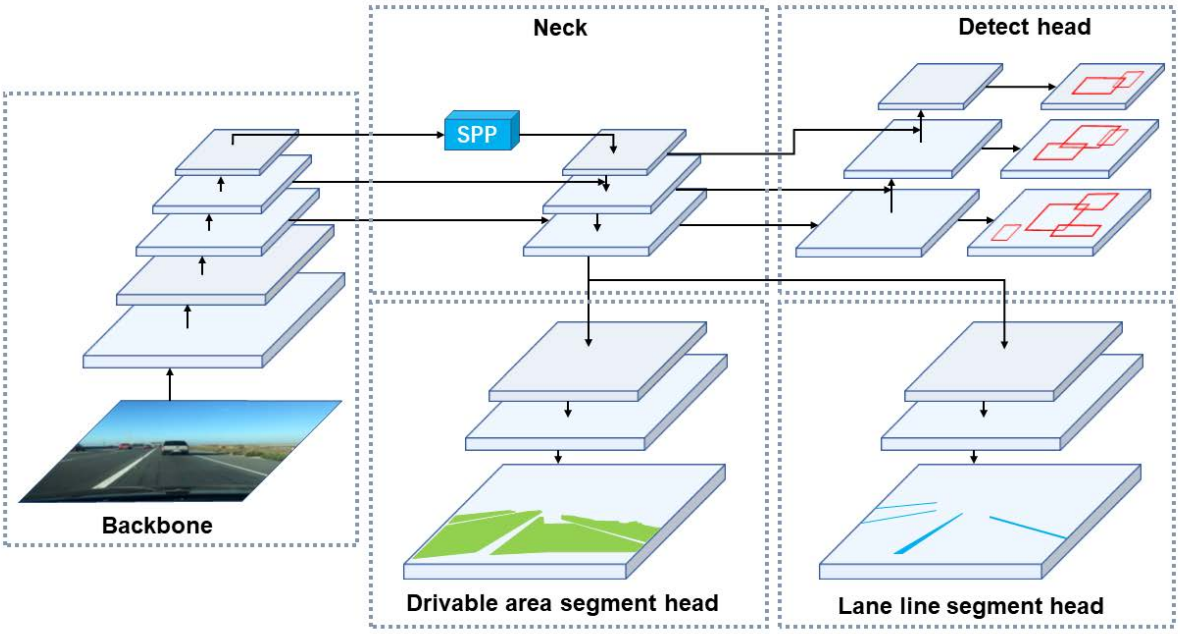


Figure 2. The architecture of YOLOP. YOLOP shares one encoder and combines three decoders to solve different tasks. The encoder consists of a backbone and a neck.

Aggregation Network (PAN), a bottom-up feature pyramid network [13]. FPN transfers semantic features top-down, and PAN transfers positioning features bottom-up. We combine them to obtain a better feature fusion effect, and then directly use the multi-scale fusion feature maps in the PAN for detection. Then, each grid of the multi-scale feature map will be assigned three prior anchors with different aspect ratios, and the detection head will predict the offset of position and the scaling of the height and width, as well as the corresponding probability of each category and the confidence of the prediction.

3.2.2 Drivable Area Segment Head & Lane Line Segment Head

Drivable area segment head and Lane line Segment head adopt the same network structure. We feed the bottom layer of FPN to the segmentation branch, with the size of $(W/8, H/8, 256)$. Our segmentation branch is very simple. After three upsampling processes, we restore the output feature map to the size of $(W, H, 2)$, which represents the probability of each pixel in the input image for the drivable area/lane line and the background. Because of the shared SPP in the neck network, we do not add an extra SPP module to segment branches like others usually do [30], which brings no improvement to the performance of our network. Additionally, we use the Nearest Interpolation method in our upsampling layer to reduce computation cost instead of deconvolution. As a result, not only do our segment decoders gain high precision output, but also be very fast during inference.

3.3. Loss Function

Since there are three decoders in our network, our multi-task loss contains three parts. As for the detection loss $\mathcal{L}_{det}$, it is a weighted sum of classification loss, object loss and bounding box loss as in equation 1.

$$\mathcal{L}_{det} = \alpha_1 \mathcal{L}_{class} + \alpha_2 \mathcal{L}_{obj} + \alpha_3 \mathcal{L}_{box}, \tag{1}$$

where $\mathcal{L}_{class}$ and $\mathcal{L}_{obj}$ are focal loss [12], which is utilized to reduce the loss of well-classified examples, thus forces the network to focus on the hard ones. $\mathcal{L}_{class}$ is used for penalizing classification and $\mathcal{L}_{obj}$ for the confidence of one prediction. $\mathcal{L}_{box}$ is $\mathcal{L}_{CIoU}$ [31], which takes distance, overlap rate, the similarity of scale and aspect ratio between the predicted box and ground truth into consideration.

Both of the loss of drivable area segmentation $\mathcal{L}_{da-seg}$ and lane line segmentation $\mathcal{L}_{ll-seg}$ contain Cross Entropy Loss with Logits $\mathcal{L}_{ce}$, which aims to minimize the classification errors between pixels of network outputs and the targets. It is worth mentioning that IoU loss: $\mathcal{L}_{IoU} =$

$1 - \frac{TP}{TP+FP+FN}$ is added to $\mathcal{L}_{ll-seg}$ as it is especially efficient for the prediction of the sparse category of lane lines. $\mathcal{L}_{da}$ and $\mathcal{L}_{ll-seg}$ are defined as equation (2), (3) respectively.

$$\mathcal{L}_{da-seg} = \mathcal{L}_{ce}, \tag{2}$$

$$\mathcal{L}_{ll-seg} = \mathcal{L}_{ce} + \mathcal{L}_{IoU}. \tag{3}$$

In conclusion, our final loss is a weighted sum of the three parts all together as in equation (4).

$$\mathcal{L}_{all} = \gamma_1 \mathcal{L}_{det} + \gamma_2 \mathcal{L}_{da-seg} + \gamma_3 \mathcal{L}_{ll-seg}, \tag{4}$$

where $\alpha_1, \alpha_2, \alpha_3, \gamma_1, \gamma_2, \gamma_3$ can be tuned to balance all parts of the total loss.

3.4. Training Paradigm

We attempt different paradigms to train our model. The simplest one is training end to end, and then three tasks can be learned jointly. This training paradigm is useful when all tasks are indeed related. In addition, some alternating optimization algorithms also have been tried, which train our model step by step. In each step, the model can focus on one or multiple related tasks regardless of those unrelated. Even if not all tasks are related, our model can still learn adequately on each task with this paradigm. And Algorithm 1 illustrates the process of one step-by-step training method.

Algorithm 1 One step-by-step Training Method. First, we only train Encoder and Detect head. Then we freeze the Encoder and Detect head as well as train two Segmentation heads. Finally, the entire network is trained jointly for all three tasks.

Input: Target neural network $\mathcal{F}$ with parameter group: $\Theta = \{\theta_{enc}, \theta_{det}, \theta_{seg}\}$;
Training set: $\mathcal{T}$;
Threshold for convergence: thr;
Loss function: $\mathcal{L}_{all}$
Output: Well-trained network: $\mathcal{F}(\mathbf{x}; \Theta)$

procedure TRAIN($\mathcal{F}$, $\mathcal{T}$)
  repeat
    Sample a mini-batch $(\mathbf{x}_s, \mathbf{y}_s)$ from training set $\mathcal{T}$.
    $\ell \leftarrow \mathcal{L}_{all}(\mathcal{F}(\mathbf{x}_s; \Theta), \mathbf{y}_s)$
    $\Theta \leftarrow \arg\min_{\Theta} \ell$
  until $\ell < thr$
end procedure
$\Theta \leftarrow \Theta \setminus \{\theta_{seg}\}$ // Freeze parameters of two Segmentation heads.
TRAIN($\mathcal{F}$, $\mathcal{T}$)
$\Theta \leftarrow \Theta \cup \{\theta_{seg}\} \setminus \{\theta_{det}, \theta_{enc}\}$ // Freeze parameters of Encoder and Detect head and activate parameters of two Segmentation heads.
TRAIN($\mathcal{F}$, $\mathcal{T}$)
$\Theta \leftarrow \Theta \cup \{\theta_{det}, \theta_{enc}\}$ // Activate all parameters of the neural network.
TRAIN($\mathcal{F}$, $\mathcal{T}$)
return Trained network $\mathcal{F}(\mathbf{x}; \Theta)$

4. Experiments

4.1. Setting

4.1.1 Dataset Setting

The BDD100K dataset [28] supports the research of multi-task learning in the field of autonomous driving. With 100k frames of pictures and annotations of 10 tasks, it is the largest driving video dataset. As the dataset has the diversity of geography, environment, and weather, the algorithm trained on the BDD100k dataset is robust enough to migrate to a new environment. Therefore, we choose the BDD100k dataset to train and evaluate our network. The BDD100K dataset has three parts, training set with 70K images, validation set with 10K images, and test set with 20K images. Since the label of the test set is not public, we evaluate our network on the validation set.

4.1.2 Implementation Details

In order to enhance the performance of our model, we empirically adopt some practical techniques and methods of data augmentation.

With the purpose of enabling our detector to get more prior knowledge of the objects in the traffic scene, we use the k-means clustering algorithm to obtain prior anchors from all detection frames of the dataset. We use Adam as the optimizer to train our model and the initial learning rate, β_1, and β_2 are set to be 0.001, 0.937, and 0.999 respectively. Warm-up and cosine annealing are used to adjust the learning rate during the training, which aim at leading the model to converge faster and better [16].

We use data augmentation to increase the variability of images so as to make our model robust in different environments. Photometric distortions and geometric distortions are taken into consideration in our training scheme. For photometric distortions, we adjust the hue, saturation and value of images. We use random rotating, scaling, translating, shearing, and left-right flipping to process images to handle geometric distortions.

4.1.3 Experimental Setting

We select some excellent multi-task networks and networks that focus on a single task to compare with our network. Both MultiNet and DLT-Net handle multiple panoptic driving perception tasks, and they have achieved great performance in object detection and drivable area segmentation

tasks on the BDD100k dataset. Faster-RCNN is an outstanding representative of the two-stage object detection network. YOLOv5 is the single-stage network that achieves state-of-the-art performance on the COCO dataset. PSPNet achieves splendid performance on semantic segmentation task with its superior ability to aggregate global information. We retrain the above networks on the BDD100k dataset and compare them with our network on object detection and drivable area segmentation tasks. Since there is no suitable existing multi-task network that processes lane detection task on the BDD100K dataset, we compare our network with Enet [19], SCNN and Enet-SAD, three advanced lane detection networks. Besides, the performance of the joint training paradigm is compared with alternating training paradigms of many kinds. Moreover, we compare the accuracy and speed of our multi-task model trained to handle multiple tasks with the one trained to perform a specific task. Furthermore, we compare the performance of semantic segmentation task combined with single-stage detection task and two-stage detection task. Following [9], we resize images in BDD100k dataset from 1280×720×3 to 640×384×3. All control experiments follow the same experimental settings and evaluation metrics, and all experiments are run on NVIDIA GTX TITAN XP.

4.2. Result

In this section, we just simply train our model end to end and then compare it with other representative models on all three tasks.

4.2.1 Traffic Object Detection Result

Visualization of the traffic objects detection is shown in Figure 3. Since the Multinet and DLT-Net can only detect vehicles, we only consider the vehicle detection results of five models on the BDD100K dataset. As shown in Table 1, we use Recall and mAP50 as the evaluation metric of detection accuracy. Our model exceeds Faster R-CNN, MultiNet, and DLT-Net in detection accuracy, and is comparable to YOLOv5s that actually uses more tricks than ours. Moreover, our model can infer in real time. YOLOv5s is faster than ours because it does not have the lane line segment head and drivable area segment head.

Figure 4 shows the qualitative comparison between Faster R-CNN and YOLOP. Due to the information share of multi-task, the prediction results of YOLOP are more reasonable. For example, YOLOP will not misidentify the objects far from the road as vehicle. Moreover, the examples of false negative are much less and the bounding boxes are more accurate.

Network	Recall(%)	mAP50(%)	Speed(fps)
MultiNet	81.3	60.2	8.6
DLT-Net	89.4	68.4	9.3
Faster R-CNN	81.2	64.9	8.8
YOLOv5s	86.8	77.2	82
YOLOP (ours)	89.2	76.5	41

Table 1. Traffic Object Detection Results: comparing the proposed YOLOP with state-of-the-art detectors.

Network	mIoU(%)	Speed(fps)
MultiNet	71.6	8.6
DLT-Net	71.3	9.3
PSPNet	89.6	11.1
YOLOP (ours)	91.5	41

Table 2. Drivable Area Segmentation Results: Comparing the proposed YOLOP with state-of-the-art drivable area segmentation or semantic segmentation methods.

4.2.2 Drivable Area Segmentation Result

Visualization results of the drivable area segmentation can be seen in Figure 5. In this paper, both “area/drivable” and “area/alternative” classes in BDD100K dataset are categorized as ”Drivable area” without distinction. Our model only needs to distinguish the drivable area and the background in the image. mIoU is used to evaluate the segmentation performance of different models. The results are shown in Table 2. It can be seen that our model outperforms MultiNet, DLT-Net and PSPNet by 19.9%, 20.2%, and 1.9%, respectively. Furthermore, our inference speed is 4 to 5 times faster than theirs.

The comparison between results of PSPNet and YOLOP is showed in Figure 6. Both PSPNet and YOLOP have perfomed well in this task. But YOLOP is significantly better at segmenting the edge areas that next to vehicles or lane lines. We think it’s mainly because that both two other tasks provide the edge information for this task. Meanwhile, YOLOP makes fewer stupid mistakes, such as misjudging the opposite lane area as drivable area.

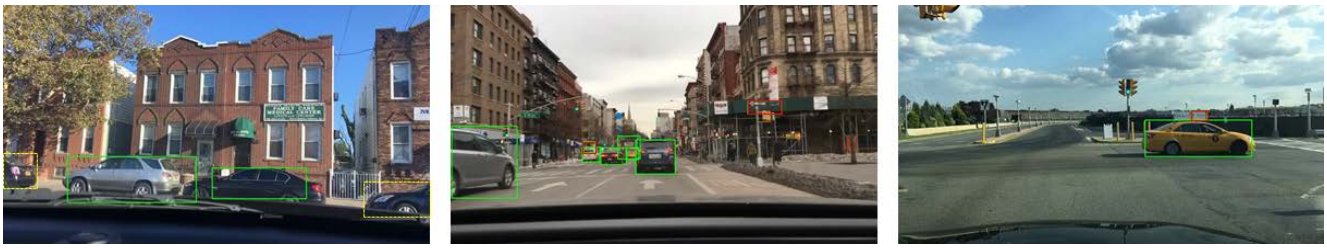
4.2.3 Lane Detection Result

The visualization results of lane detection can be seen in Figure 7. The lane lines in BDD100K dataset are labeled with two lines, so it is very tricky to directly use the annotation. The experimental settings follow the [9] in order to compare expediently. First of all, we calculate the center lines based on the two-line annotations. Then we draw

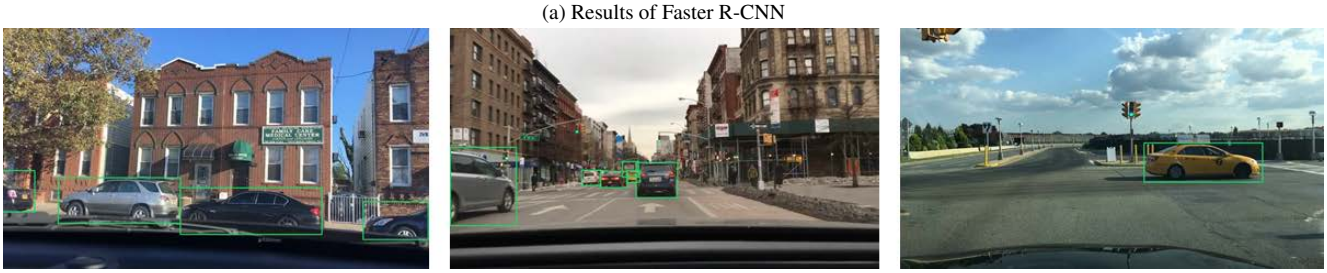
(a) Day-time result

(b) Night-time result

Figure 3. Visualization of the traffic objects detection results of YOLOP. Top Row: Traffic objects detection results in day-time scenes. Bottom row: Traffic objects detection results in night scenes.



(a) Results of Faster R-CNN



(b) Results of YOLOP

Figure 4. Comparison between the traffic objects detection results of Faster R-CNN and YOLOP. Top Row: Traffic objects detection results of Faster R-CNN. Bottom row: Traffic objects detection results of YOLOP. The green bounding boxes are the detected correct vehicles. The yellow dotted bounding boxes are the false negative. The red bounding boxes indicate the false positive.

the lane line of the training with width set to 8 pixels while keeping the lane line width of the test set as 2 pixels. We use pixel accuracy and IoU of lanes as evaluation metrics. As shown in the Table 3, the performance of our model dramatically exceeds the other three models.

Figure 8 shows the comparison of Lane line detection results of ENet-SAD and YOLOP. The segmentation results of YOLOP is more accurate and continuous than ENet-SAD obviously. With the imformation shared by the other two tasks, YOLOP will not mistake some areas where some vehicles are located or driveable as lane lines, but Enet-SAD always does.

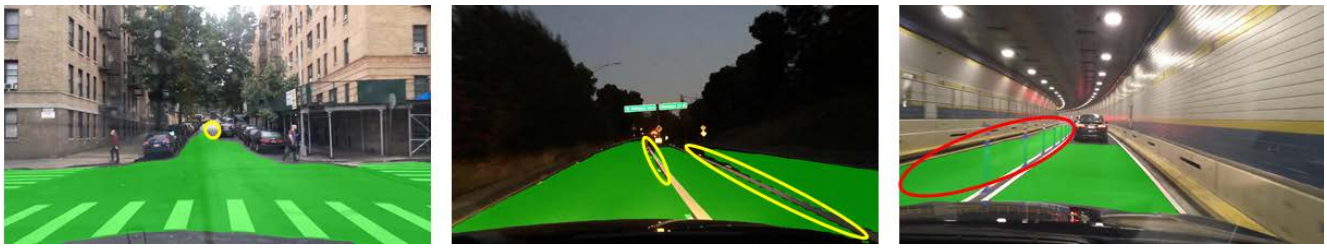
4.3. Ablation Studies

We designed the following three ablation experiments to further illustrate the effectiveness of our scheme. All the evaluation metrics in this section are consistent with above.

(a) Day-time result

(b) Night-time result

Figure 5. Visualization of the drivable area segmentation results of YOLOP. Top Row: Drivable area segmentation results in day-time scenes. Bottom row: Drivable area segmentation results in night scenes.



(a) Results of PSPNet

(b) Results of YOLOP

Figure 6. Comparison between the drivable area segmentation results of PSPNet and YOLOP. Top Row: Drivable area segmentation results of PSPNet. Bottom row: Drivable area segmentation results of YOLOP. The yellow ellipses are the false negative. The red ellipses indicate the false positive.

4.3.1 End-to-end v.s. Step-by-step

In Table 4, we compare the performance of joint training paradigm with alternating training paradigms of many kinds [1]. Obviously, our model has performed very well enough through end-to-end training, so there is no need to perform alternating optimization. However, it is interesting that the paradigm training detection task firstly seems to perform better. We think it is mainly because our model is closer to a complete detection model and the model is harder to converge when performing detection tasks. What's more, the paradigm consist of three steps slightly outperforms that with two steps. Similar alternating training can be run for more steps, but we have observed negligible improvements.

[1] E, D, S and W refer to Encoder, Detect head, two Segment heads and whole network. So the Algorithm 1 can be marked as ED-S-W, and the same for others.

(a) Day-time result

(b) Night-time result

Figure 7. Visualization of the lane detection results of YOLOP. Top Row: Lane detection results in day-time scenes. Bottom row: Lane detection results in night scenes.

(a) Results of ENet-SAD

(b) Results of YOLOP

Figure 8. Comparison between the lane detection results of ENet-SAD and YOLOP. Top Row: Lane detection results of ENet-SAD. Bottom row: Lane detection results of YOLOP. The yellow ellipses are the false negative. The red ellipses indicate the false positive.

4.3.2 Multi-task v.s. Single task

To verify the effectiveness of our multi-task learning scheme, we compare the performance of the multi-task scheme and single task scheme. On the one hand, we train our model to perform 3 tasks simultaneously. On the other hand, we train our model to perform traffic object detection, drivable area segmentation, and lane line segmentation tasks separately. Table 5 shows the comparison of the performance of these two schemes on each specific task. It can be seen that our model adopts the multi-task scheme to achieve performance is close to that of focusing on a single task. More importantly, the multitask model can save a lot of time compared to executing each task individually.

4.3.3 Region-based v.s. Grid-based

To verify the viewpoint that the grid-based prediction mechanism is more related to the two semantic segmentation tasks than the region-based prediction mechanism. We extend Faster R-CNN by adding two semantic segment heads to perform three tasks in parallel as our model did, and we

Network	Accuracy(%)	IoU(%)	Speed(fps)
ENet	34.12	14.64	100
SCNN	35.79	15.84	19.8
ENet-SAD	36.56	16.02	50.6
YOLOP (ours)	70.50	26.20	41

Table 3. Lane Detection Results: comparing the proposed YOLOP with state-of-the-art lane detection methods.

call such a new model R-CNNP. We train both YOLOP and R-CNNP to (i) perform detection task and two segmentation tasks separately and (ii) three tasks simultaneously. In both two experiments above, the two segmentation tasks are trained jointly as there is no need to consider the interaction between them. All the experimental settings are the same, and the results are shown in Table 6. In the R-CNNP framework, the performance of multi-task training is much worse compared with training the detection task and semantic segmentation tasks separately. Obviously, the combination of two kinds of tasks conflicts in R-CNNP framework. But there is no such problem in our YOLOP framework, the performance of multi-task training is equal to that of focusing only on detection or semantic segmentation task. Thus we hold the opinion that this is due to the detection head of YOLOP, like two other semantic segmentation heads, directly perform global classification or regression tasks on the whole feature map output by Encoder, so they are similar and related in terms of prediction mechanism. Nevertheless, the detection head of R-CNNP needs to select region proposals first, and then perform prediction on the feature maps of each individual proposals, which is quite different from the global prediction mechanism of semantic segmentation. In addition, R-CNNP is far behind YOLOP in terms of inference speed. Therefore, our framework is a better choice for joint training detection and segmentation tasks.

5. Conclusion

In this paper, we put forward a brand-new, simple and efficient network, which can simultaneously handle three driving perception tasks of object detection, drivable area segmentation and lane detection and can be trained end-to-end. Our model performs exceptionally well on the challenging BDD100k dataset, achieving or greatly exceeding state-of-the-art level on all three tasks. And it is the first to realize real-time reasoning on embedded device Jetson TX2, which ensures that our network can be used in real-world scenarios. Moreover, we have verified that the grid-based prediction mechanism is more related to that of semantic segmentation task. which may be of certain reference significance to similar multi-task learning research works.

Currently, although our multi-task network can be trained end-to-end without compromising the performance of each other, we hope to improve the performance of those tasks with more appropriate paradigm for multitask learning. Furthermore, our model is limited in three tasks, more tasks related with autonomous driving perception system such as depth estimation can be added in our future frame work to make the whole system more complete and practical.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020. 2, 3

[2] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. arXiv preprint arXiv:1605.06409, 2016. 2

[3] Kaiwen Duan, Lingxi Xie, Honggang Qi, Song Bai, Qingming Huang, and Qi Tian. Location-sensitive visual recognition with cross-iou loss. arXiv preprint arXiv:2104.04899, 2021. 3

[4] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 1440–1448, 2015. 2

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014. 2

[6] Hsiang-Yu Han, Yu-Chi Chen, Pei-Yung Hsiao, and Li-Chen Fu. Using channel-wise attention for deep cnn based real-time semantic segmentation with class-aware edge information. IEEE Transactions on Intelligent Transportation Systems, 22(2):1041–1051, 2020. 3

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 2961–2969, 2017. 2, 3

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9):1904–1916, 2015. 3

[9] Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection cnns by self attention distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1013–1021, 2019. 2, 3, 6

[10] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In ICML, 2011. 3

[11] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2117–2125, 2017. 3

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 2980–2988, 2017. 4

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018. 4

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37. Springer, 2016. 2

[15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015. 2

[16] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983, 2016. 5

[17] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In 2018 IEEE Intelligent Vehicles Symposium (IV), pages 286–291. IEEE, 2018. 3

[18] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018. 2, 3

[19] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147, 2016. 2, 3, 6

[20] Yeqiang Qian, John M Dolan, and Ming Yang. Dlt-net: Joint detection of drivable areas, lane lines, and traffic objects. IEEE Transactions on Intelligent Transportation Systems, 21(11):4670–4679, 2019. 3

[21] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 779–788, 2016. 2

[22] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7263–7271, 2017. 2

[23] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018. 2

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497, 2015. 2, 3

[25] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. Multinet: Real-time joint semantic reasoning for autonomous driving. arXiv preprint arXiv:1612.07695, 2016. 3

[26] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. arXiv preprint arXiv:2011.08036, 2020. 2, 3

[27] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. Cspnet: A new backbone that can enhance learning capability of cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 390–391, 2020. 3

[28] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. arXiv preprint arXiv:1805.04687, 2(5):6, 2018. 2, 5

[29] Jie Zhang, Yi Xu, Bingbing Ni, and Zhenyu Duan. Geometric constrained joint lane segmentation and lane boundary detection. In Proceedings of the European Conference on Computer Vision (ECCV), pages 486–502, 2018. 3

[30] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017. 2, 3, 4

[31] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 12993–13000, 2020. 4

Training method	Recall(%)	AP(%)	mIoU(%)	Accuracy(%)	IoU(%)
ES-W	87.0	75.3	90.4	66.8	26.2
ED-W	87.3	76.0	91.6	71,2	26.1
ES-D-W	87.0	75.1	91.7	68.6	27.0
ED-S-W	87.5	76.1	91.6	68.0	26.8
End-to-end	89.2	76.5	91.5	70.5	26.2

Table 4. Panoptic driving perception results: the end-to-end scheme v.s. different step-by-step schemes.

Training method	Recall(%)	AP(%)	mIoU(%)	Accuracy(%)	IoU(%)	Speed(ms/frame)
Det(only)	88.2	76.9	-	-	-	15.7
Da-Seg(only)	-	-	92.0	-	-	14.8
Ll-Seg(only)	-	-	-	79.6	27.9	14.8
Multitask	89.2	76.5	91.5	70.5	26.2	24.4

Table 5. Panoptic driving perception results: multi-task learning v.s. single task learning.

Training method		Recall(%)	AP(%)	mIoU(%)	Accuracy(%)	IoU(%)	Speed(ms/frame)
R-CNNP	Det(only)	79.0	67.3	-	-	-	-
	Seg(only)	-	-	90.2	59.5	24.0	-
	Multitask	77.2(-1.8)	62.6(-4.7)	86.8(-3.4)	49.8(-9.7)	21.5(-2.5)	103.3
YOLOP	Det(only)	88.2	76.9	-	-	-	-
	Seg(only)	-	-	91.6	69.9	26.5	-
	Multitask	89.2(+1.0)	76.5(-0.4)	91.5(-0.1)	70.5(+0.6)	26.2(-0.3)	24.4

Table 6. Panoptic driving perception results: Grid-based v.s. Region-based.